Writing Assignment 3

Haniah Kring

2023-10-22

1. Using initial state $\begin{bmatrix} P_0 \\ Q_0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$, fill out the following table.

n	0	1	2	3	4	5	6	7	8
P_n	3								
Q_n	1								

A matrix dynamical system allows us to start from the initial state and forecast the population model at any future year. If $A = \begin{bmatrix} 8 & -3 \\ 18 & -7 \end{bmatrix}$ and $\begin{bmatrix} P_0 \\ Q_0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$, then

$$\begin{bmatrix} P_1 \\ Q_1 \end{bmatrix} = A \begin{bmatrix} P_0 \\ Q_0 \end{bmatrix} = \begin{bmatrix} 21 \\ 47 \end{bmatrix}$$

$$\begin{bmatrix} P_2 \\ Q_2 \end{bmatrix} = A^2 \begin{bmatrix} P_1 \\ Q_1 \end{bmatrix} = \begin{bmatrix} 27 \\ 49 \end{bmatrix}$$

$$\vdots$$

$$\begin{bmatrix} P_8 \\ Q_8 \end{bmatrix} = A^8 \begin{bmatrix} P_7 \\ Q_7 \end{bmatrix} = \begin{bmatrix} 2043 \\ 4081 \end{bmatrix}$$

```r
A <- matrix(nrow = 2, ncol = 2, data = c(8, -3, 18, -7), byrow = TRUE)
v <- c(3, 1)
n <- 0:8

result_list <- sapply(n, function(x) {
    (A %^% x) %*% v
})
```

```r
colnames(result_list) <- n
rownames(result_list) <- c("P~n~", "Q~n~")

knitr::kable(result_list)
```

	0	1	2	3	4	5	6	7	8
P_n	3	21	27	69	123	261	507	1029	2043
Q_n	1	47	49	143	241	527	1009	2063	4081

2. Add to your table a row for $T_n = P_n + Q_n$.

```r
Tn <- sapply(n+1, function(x) {
  result_list[[1, x]] + result_list[[2, x]]
})

result_list <- rbind(result_list, Tn)

rownames(result_list) <- c("P~n~", "Q~n~", "T~n~")

knitr::kable(result_list)
```

	0	1	2	3	4	5	6	7	8
P_n	3	21	27	69	123	261	507	1029	2043
Q_n	1	47	49	143	241	527	1009	2063	4081
T_n	4	68	76	212	364	788	1516	3092	6124

3. Use your data to predict the value of $\lim_{n\to\infty} \frac{T_n+1}{T_n}$.

Based off the data, we can observe that as n approaches infinity, $\lim_{n\to\infty} \frac{T_n+1}{T_n} = 2$.

4. Add to your table a row for $R_n = \frac{Q_n}{P_n}$.

```r
Rn <- sapply(n+1, function(x) {
  round(result_list[[2, x]] / result_list[[1, x]], 3)
})

result_list <- rbind(result_list, Rn)

rownames(result_list) <- c("P~n~", "Q~n~", "T~n~", "R~n~")
```

```r
knitr::kable(result_list)
```

	0	1	2	3	4	5	6	7	8
P_n	3.000	21.000	27.000	69.000	123.000	261.000	507.00	1029.000	2043.000
Q_n	1.000	47.000	49.000	143.000	241.000	527.000	1009.00	2063.000	4081.000
T_n	4.000	68.000	76.000	212.000	364.000	788.000	1516.00	3092.000	6124.000
R_n	0.333	2.238	1.815	2.072	1.959	2.019	1.99	2.005	1.998

5. Use your data to predict the value of $\lim_{n\to\infty} R_n$.

Based off the data, we can observe that as n approaches infinity, $\lim_{n\to\infty} R_n = 2$ as well.

6. Repeat #1 through #5 with initial state $\begin{bmatrix} P_0 \\ Q_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. What is the effect (if any) on your answers to #3 and #5?

```r
v <- c(1, 1)

result_list <- sapply(n, function(x) {
    (A %^% x) %*% v
})

Tn <- sapply(n+1, function(x) {
  result_list[[1, x]] + result_list[[2, x]]
})

result_list <- rbind(result_list, Tn)

Rn <- sapply(n+1, function(x) {
  round(result_list[[2, x]] / result_list[[1, x]], 3)
})

result_list <- rbind(result_list, Rn)

colnames(result_list) <- n
rownames(result_list) <- c("P~n~", "Q~n~", "T~n~", "R~n~")

knitr::kable(result_list)
```

	0	1	2	3	4	5	6	7	8
P_n	1	5.0	7.000	17.000	31.000	65.000	127.000	257.000	511.000
Q_n	1	11.0	13.000	35.000	61.000	131.000	253.000	515.000	1021.000
T_n	2	16.0	20.000	52.000	92.000	196.000	380.000	772.000	1532.000
R_n	1	2.2	1.857	2.059	1.968	2.015	1.992	2.004	1.998

The only effect is that I am now even more confident the limit is 2. As n increases, R_n approaches 2.

7. What happens when the initial state vector is $\begin{bmatrix} P_0 \\ Q_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$? Here, determine exact formulas for $\begin{bmatrix} P_n \\ Q_n \end{bmatrix}$.

```r
v <- c(1, 2)

result_list <- sapply(n, function(x) {
    (A %^% x) %*% v
})

Tn <- sapply(n+1, function(x) {
  result_list[[1, x]] + result_list[[2, x]]
})

result_list <- rbind(result_list, Tn)

Rn <- sapply(n+1, function(x) {
  round(result_list[[2, x]] / result_list[[1, x]], 3)
})

result_list <- rbind(result_list, Rn)

colnames(result_list) <- n
rownames(result_list) <- c("P~n~", "Q~n~", "T~n~", "R~n~")

knitr::kable(result_list)
```

	0	1	2	3	4	5	6	7	8
P_n	1	2	4	8	16	32	64	128	256
Q_n	2	4	8	16	32	64	128	256	512

	0	1	2	3	4	5	6	7	8
T_n	3	6	12	24	48	96	192	384	768
R_n	2	2	2	2	2	2	2	2	2

When the initial state vector is $\begin{bmatrix} P_0 \\ Q_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, each value for P_0 and Q_0 doubles on each iteration. Exact formula: $\begin{bmatrix} P_n \\ Q_n \end{bmatrix} = \begin{bmatrix} 2^i \\ 2(2^i) \end{bmatrix}$. The pattern for R_n is self evident. The 2^i in the numerator and denominator cancel out, leaving only 2 in the numerator.

8. Write a summary paragraph that gives your conclusions or predictions based on your answers above.

Based off the results of these calculations, we can reasonably conclude that no matter what initial state vector you pick, the limit will always approach 2. The initial state only determines how quickly the limit is reached. How the model functions then is a result of the initial matrix you choose, and as the model progresses, it averages out to a particular value.